Reasons why end-users abandon or neglect features in end-user programming environments – Term Case Study Project (CS569)

Executive Summary

1. Research question	When and why do end-user programmers give up on learning/using any feature/functionality related to the end-user programming environment?
2. Propositions	**Proposition1**: Strategies adopted: exploratory learning, help of colleagues, internet, online forums, documentation, the embedded help system, or library. **Proposition2**: The main reasons for selecting a strategy/or for giving up– may be tied to the help system, the time, confidence, gender, background (age, their education), curiosity, level of expertise, dire necessity to use a feature, frustration, motivation, interest (interest in learning, versus wanting to accomplish the task at hand), complexity of the task, resistance to adopt new features, past negative experiences, poorly designed system etc. **Proposition3**: Six learning barriers that might contribute to their abandonment – Design, selection, coordination, use, understanding and information
3. Type of case study	Explanatory
4a. Units of analysis and units of observation.	**Unit of analysis**: End-user programmer. Our unit of analysis is our primary unit of observation as well.
4b. Data collected.	- Direct observation of the end-user programmers - Direct observation of the training session given to the end-user programmers (to understand the level of training the users had) - Semi structured interviews with the end-user programmers - Questionnaires to the end-user programmers - Documentation of the software - The beta version of the software (for us to tinker with)
5. Design type.	Multiple Case Holistic

Introduction:

There has been a body of work on technology acceptance as such [11,13,14], identifying the reasons for resistance, with possible solutions. But, this focuses on a universal set per se, i.e, technology, but we are attempting to understand a subset and possibly a more significant aspect of software utilization – usage of features. It is a known problem that with any given software or programming environment, end-users do not necessarily adopt all the features and functionalities made available in the environment. As a matter of fact, [4] point out how even experienced users fail to realize how to use features to fully capitalize the functionalities of the software. One example they give is how most users of Excel manually resize each column, rather than selecting all the columns and then applying the resize command. Although this work focuses on optimal usage of features, it is evidence to show how regardless of experience, users bypass the need to use/learn features or most of the functionalities of the software to apply to their task, and instead would choose to manually perform the task. This behavior of users has been discussed by the paradox of the active user [5] and minimalist theory [6] that learning is somewhat counter to the goal of completing the task, because it takes away time from the task. This aspect is also echoed by Blackwell's model of attention investment [3], which shows how users weigh the attention cost of learning something new against the risk that the learning effort will not cause time savings.

Although developers of end-user programming environments have started to pay attention to these theories while designing, there are still some open ended questions on what are the main reasons that lead to users' abandonment of a feature. Reluctance to learn [5], learning barriers [9], self-efficacy [1], gender differences [2], poor design guidelines mainly with help systems [7] are some reasons that have been put forth by past research. While this definitely sheds light on the main reasons that might cause abandonment of features/functionality, it is still not clear, *how* these reasons affect one another to lead to barriers that cannot be overcome. For example, we are still not sure *what* causes learning barriers, invalid assumptions [9] and *why* they cannot be overcome. We are attempting to *explain* the cause-effect relationship between the various factors that have been identified, and the *actual behavior* of the end-users, in a *real-world* context.

Towards this, our main research question is – *Why and when do end-user programmers give up on learning/using any feature/functionality related to the end-user programming environment.* Further refining this broad question, we attempt at explaining the following questions-

RQ1: *What are the strategies end-user programmers adopt, to resort to help/clarify their understanding on any aspect of the programming environment. Why do they perceive it as being ideal for them?*

Proposition1: exploratory learning, help of colleagues, internet, online forums, documentation, the embedded help system, or library [12].

RQ2*: How many features/functionalities have been unused or neglected in the environment and why?*

RQ3: *When/Why are the end-users successful with a feature/functionality?*

Proposition2/3: May be tied to the help system [7], the time [3], self-efficacy [1], gender [2, 14], background (age, their education), curiosity [10], level of expertise, dire necessity to use a feature, frustration, motivation, interest (interest in learning, versus wanting to accomplish the task at hand), complexity of the task, resistance to adopt new features [11, 2], past negative experiences, poorly designed system etc.

RQ4*: What learning barriers were identified?*

Proposition4: Six learning barriers – Design, selection, coordination, use, understanding and information [9]

Design of the study:

Case/Unit of analysis: The unit of analysis that best addresses the research question is the end-user programmer. We will be studying beginner/novice end-user programmers' usage of the i5Logic Audit Beta software, embedded within Microsoft Excel. Therefore, the environment is Excel with the addition of embedded features from the Audit software. The participants will be end-users (with minimal computer literacy) within the OSU community. The participants of our case study fall under the category of "active users" [5] whose goal is *not* to learn the software, but to use it to get their task done, and this is the audience we are particularly interested in. Our unit of observation is also the end-user programmer.

The Audit software embedded in Excel, presents a set of features that employs the WYSIWYT technology with fault localization techniques, mainly oriented to improve reliability and correctness of spreadsheets created by end-users. Therefore the set of features are more conducive to be used during the testing and debugging phases of software development. But we are also expanding the feature set of our observation to include any Excel features that they might use during the course of our observation.

Type of study: This is an explanatory case study. We are building on well defined theories and significant past research to enrich our understanding on what causes end-users to neglect or give up using features in the end-user programming environment. Our main goal is to *explain* cause-effect relationships between the reasons identified by theories that could contribute to users' abandonment of features/functionalities, and *how* the end-users' behavior thus gets affected in a real-world scenario. We are trying to explain how the theories apply to end-users in the creation/maintenance/testing phases of software development.

Design Type: This is a multiple case holistic study. We will be observing more than one end-user programmer (our case). We view each case as being distinct/unique, since we deal with human subjects. We also intend to perform cross case analysis, which can only be done across multiple cases. This is a holistic study because there are no sub units of analysis. The software and other factors are variables that we measure in order to understand the main unit.

Data to be collected:

- Direct observation of the end-user programmers – To understand their actual behavior and responses to the system and the reasons that cause them to respond so.
- Direct observation of the training session – To understand the level of training the end-users go through prior to using the Audit software in Excel.
- Semi structured interviews with the end-user programmers – To clarify their notion of the software, its usability, reasons for their behavior and other surprises that we might encounter, while observing and during the course of the interview.
- Questionnaires to the end-user programmers – Such as self-efficacy, background to formally measure and understand contributing factors.
- Documentation of the software – in order for us to understand the software and to understand how effective/useful the documentation is.
- The beta version of the software – for us to get familiarized with the features and the functionalities of the software in order to better understand what to look for while observing, and what we need to measure.

Natural Controls: We are planning to observe four participants (cases). An even distribution of gender is not a control we enforce while recruiting our participants. But if it so happens that we have only males or only females as our participants, it would limit our analysis to some extent. But if we get an even distribution of males and females, our analysis and explanations would be highly strengthened, only if we choose to consider this opportunity. Another example would be, if a participant uses only Excel's features during the first half of the observation, and only the features from i5Logic during the second half, it would help us make a clear distinction on their behavioral pattern with Excel versus the new product.

Case Study Protocol

Overview of the case study project:

- ***Objectives:*** The intent of the project is to understand the behavior of beginner/novice end-user programmers towards a newly introduced set of software functionalities, embedded in Microsoft Excel. The environment is Excel with the addition of embedded features from i5Logic's Audit software. More specifically our aim is to understand the reasons behind end-users' abandonment or negligence of the features and functionalities available in the programming environment. The goal of this project is to explain what causes end-users to reach a stage where they give up on learning/using a feature. Therefore our objective is to pay attention to, and be sensitive to all possible issues/reasons (those identified by theories, and those that are observed in the field) that might contribute to the effect we are trying to explain – abandonment of features. Thus the research question that is to be addressed by this project is: *Why and when do end-user programmers give up on learning/using any feature/functionality related to the end-user programming environment?*

- ***Theoretical Framework:*** The participants of our project belong to the category of 'active users' [5]. Hence the logic (predictive) model that best suits this case study, is the *'Paradox of the Active User and Minimalism theory'* [5,6]. As this theory points out, the end-users in the study are very wary of the time they need to spend away from the task, in order to learn the software. We need to bear in mind that they are likely to have very minimal interest to learn the software. Blackwell's model of *'Attention Investment'* [3], is heavily tied to this behavior of active users, because only if the perceived benefits of learning/using a feature is viewed greater than the risks, will they consider investing their time on the feature worthwhile. But another angle to consider is the barriers users face while learning the software. The six learning barriers identified by [9] will serve as good formal models for us to pattern match our observations and findings.

- ***Relevant Readings:*** Reluctance to learn [5], learning barriers [9], self-efficacy [1], gender differences [2], poor design guidelines mainly with help systems [7], motivation [11,13,14], time factor [3], lack of curiosity [10] are some reasons that have been attributed to users' negligent behavior towards features. In addition to these, our propositions heavily rely on past research, as indicated in the next section.

- ***Propositions/hypothesis:***

Question1: *What are the strategies end-user programmers adopt, to resort to help/clarify their understanding on any aspect of the programming environment. Why do they perceive it as being ideal for them?*

Proposition1: exploratory learning, help of colleagues, internet, online forums, documentation, the embedded help system, or library [12].

Question2*: How many features/functionalities have been unused or neglected in the environment and why?*
Question3: *When/Why are the end-users successful with a feature/functionality?*
Proposition2/3: May be tied to the help system [7], the time [3], self-efficacy [1], gender [2, 14], background (age, their education), curiosity [10], level of expertise, dire necessity to use a feature, frustration, motivation, interest (interest in learning, versus wanting to accomplish the task at hand), complexity of the task, resistance to adopt new features [11, 2], past negative experiences, poorly designed system etc.
Question4*: What learning barriers were identified?*
Proposition4: Six learning barriers – Design, selection, coordination, use, understanding and information [9]

- ***Auspices:*** Our case study project will have participants from the Oregon State University's community, spanning across business managers, professors and accountants. The participants will be using the beta version of i5Logic's, Audit software, embedded in Microsoft Excel.

- ***Case Study Issues***:

Issue1: Our main question is why and when do users give up on using a feature. But it is a possibility that we might never/hardly encounter the exact situation of users slogging through a feature and then giving up on it. Observing a user sail smoothly through the software is not necessarily an indication that they have no problems with the features available, and they would resort to them when needed. We need to understand how many features in the feature set, are regularly used by the participants. This will help us to recognize the pattern of usage of features – for example, how many basic features are regularly used, to bypass usage of more complex features, how many features are unknown to the users, how many were previously used and abandoned etc.

Issue2: In order to get at the questions arising from issue1, the software needs to have a substantial feature set (having more than 2 features for example). Only then would it strengthen our analysis. Currently the Audit software does not have an extensive feature set (it has a set of five to six features/feedback for users to work with). Microsoft Excel on the other hand, has a much broader feature set for us to work with. Therefore Excel's features also fall under the feature set whose usage we would like to observe.

Issue3: If Excel's features are included as part of our observation and analysis, we need to stay clean in not mixing data from the participant's usage of Excel features, and Audit's features. Excel users might be more comfortable, more experienced, whereas those of Audit are fairly new. Therefore to eliminate bias, we need to tease apart data from usage of Excel features from those related to Audit.

Issue4: Our propositions include a lot of qualitative evidence that needs to be observed/collected, such as curiosity, frustration, motivation etc. It is not possible to accurately measure each of these variables, because the levels of these measures are an informed guess (as per observation and interviews), not the precise quantitative level. But nonetheless we need to be open minded and sensitive to recognize such

characteristics and record them as precise as possible (through data and investigator triangulation).

Issue5: We need to be very careful not to lead the participants while interviewing them. We also need to watch out against planting the idea that we are trying to measure and test the participants' level of knowledge with regards to using the software. This is bound to make them conscious to impress us and to do what they believe is expected out of them. This will heavily bias the data.

Issue6: Since we are observing human participants, and this is mainly a qualitative study, we need to observe and gather data that is heavily subjective. It is imperative that both the investigators be at the field site while observing, so that it gives way to investigator triangulation and catching surprises and elements that one of the observers might miss.

Issue7: While observing, ask questions only when necessary, to aid in understanding the context. But be careful not to eat into their time, by asking more follow up questions. Make a note of the questions, and ask them during the interview.

Field Procedures:

- ***Presentation of credentials:***
    - Direct observation of the participants – requires an unobtrusive but at the same time efficient recording device, such as a laptop, or notebook. In case of a notebook make sure it has plenty of pages to write on, and remember to carry extra set of pencils, pens, and erasers. If one is not too fast with writing down points, then a laptop would be ideal to type out notes and observations. While using a laptop, it serves the dual purpose of being out of the participant's way, as you could pretend to seriously work. Have a clear template to record your observations, mention all the key factors that you need to pay attention to and record (carry the database structure to take down notes). Have a clear demarcation of the raw notes and your thoughts on it. Always remember to carry your laptop's charger. Make it a practice to record/complete your notes completely after the observation. While observing, do not make the users uncomfortable – be subtle at the same time efficient in observing.
    - Direct observation of the training session – note down aspects on the level of training the users had; was it purely a lecture format, or were there demos involving hands on practice, etc.
    - Conducting interviews – While conducting interviews write down notes, carry some files (preferably one that has multiple leaves) to clearly sort out any documents or paper notes that need to be filed at the end of interview. Paper clips and post-its will be ideal to better organize the folder. It's preferable that more than one investigator is at the field while interviewing, so that question asking pattern can be alternated while the other takes down notes. Also since these are semi structured interviews we need to intelligently pick up on follow up questions and lead them on properly. Having more than one investigator ensures that such follow up questions are not missed.

- Questionnaires – make sure they are labeled with a unique id, such that they cannot be tied to the participant's real identity. If they involve color images, make sure the colors come out right, and you print out the right number of pages. Clip all pages of the questionnaire (if it involves more than one page) before handing it to the participant. Make sure to file away the filled questionnaires in an organized file.
- Documentation – Be sure to include the entire documentation of the software in the database.
- Credentials of the investigator – prior to visiting the site, the investigator needs to be equipped with the following credentials – Ability to ask good questions, and interpret their answers, good listeners not trapped by their own ideologies, adaptive and flexible to newly encountered situations, and should be able to see such situations as opportunities and not threats, must have a firm grasp on the issues being studied, unbiased by preconceived notions.
- Case study database – must be well tabulated and organized such that there is a logical flow of information, making analysis intuitive and easy.

- ***Access to case study sites:*** *Name of sites*
  The finance department of Oregon State University for observing and interviewing our four participants, office of the lead of i5Logic Audit's beta testing team, the EECS administrative office of Oregon State University (for pilot study), and the training session of Audit software.
  Calendar period for the site visit:
  Meeting up with the testing lead – May 7th 2007 (1:30 P.M-2.00 P.M -duration 30 minutes)
  Pilot subject – May 8 2007 (duration 3.00 P.M to 5.00 P.M - 2 hours)
  Training session – May 17 2007 (duration 11:00 A.M to12:00 P.M – 1 hour)
  Participant A – May 18 2007 (duration 11:00 A.M to 5 P.M- 5 hours)
  Participant B – May 21 2007 (duration 11:00 A.M to 5 P.M- 5 hours)
  Participant C – May 22 2007 (duration 11:00 A.M to 5 P.M- 5 hours)
  Participant D – May 24 2007 (duration 11:00 A.M to 5 P.M- 5 hours)
  Level of effort:
  Participant A – May 18 2007 (4 hours observation + 1 hour of interview)
  Participant B – May 21 2007 (4 hours observation + 1 hour of interview)
  Participant C – May 22 2007 (4 hours observation + 1 hour of interview)
  Participant D – May 24 2007 (4 hours observation + 1 hour of interview)

- ***General sources of information***: We need to know how experienced our participants would be. This information can be got from the head of the testing team of i5Logic. We need to review the documentation of the software, obtained from the head of i5Logic. The permission to conduct a study within OSU's finance department, must be obtained from the IRB. The permission to obtain the documents of the Audit software, and the beta version of the product itself, from the head of i5Logic. Permission to conduct our pilot study within the

administrative office of EECS department of OSU, from the participant we request.

- ***Procedural reminders (in the sequence in which it is to be executed):***

    - Conduct pilot study (to be done by May 8 2007)
    - Refine the study's design based on the pilot (if any revisions, to be done by May 14, 2007)
    - Analytically evaluate the software using cognitive dimensions [8] before observing the participants. (to be done before May 13, 2007)
    - At the site of observation, introduce yourself, state your purpose, make yourself friendly and non intimidating.
    - Establish a comfortable agreement on how to observe them working on their spreadsheet (is it ok to observe them as they work, would they rather prefer explaining what happened after they are done etc.)
    - Hand in the background questionnaire before start of observation and file it.
    - Next hand in the pre-self-efficacy questionnaire and file the filled version
    - Start observing and taking notes. Pay attention to the template so that no points are missed out.
    - Do not bug the participants with questions during observation.
    - Complete notes during short breaks that the participant takes
    - Pay specific attention to remain unobtrusive when they work on the spreadsheets.
    - Always be open to surprises and record them
    - Don't be biased. Write down actual events, in addition to your own thoughts.
    - Before the last hour, hand in the post self efficacy questionnaire and file it.
    - Let them know when you are going to interview
    - Get your recording device, to take down notes ready
    - Decide previously which investigator will ask handle which question in the predefined structure. The follow ups are dependent on the situation.
    - Start with warm up questions
    - Intelligently follow up on unanticipated questions
    - Keep a checklist on the whether the agenda of questions you wanted to ask is complete.
    - Finish the interview by thanking them
    - Immediately complete your notes.

Case Study Questions:

Level 1 Questions:

1. How would you rate the tasks you performed today in terms of their complexity?
2. Do you have any deadlines to complete the tasks you worked on today?
3. What were the potential benefits you expected from the software that made you register for a training session?
4. How useful do you think the Audit software was for your tasks?
5. How useful was the training session to help you get started?
6. How useful was the documentation?
7. I noticed you using XXX features a lot. Were they helpful? Any reasons for liking them (may be follow up)
8. Do you have any feature in Excel that you would not like to use? (follow up: Why, what are they, what would you use in substitution)
9. Do you have any feature in Audit that you would not like to use? (follow up: Why, what are they, what would you use in substitution)

Potential follow ups: if you note down any learning barrier that you would like to discuss with them, about some particular problems they were stuck, their feelings etc.

Level 2 Questions:

Question1: *What do end-users do when they want to understand a feature?*
Question2*: Among the different strategies that they use to understand a feature, which one do they prefer?*
Question3: *How favorite is the embedded help system in the environment to help clarify their understanding?*
Question4: *How many basic features were regularly used?*
Question5*: How many features were unused or bypassed, with the usage of simpler more basic features?*
Quesion6*: How many features were completely neglected due to unawareness?*
Question7*: How much time and effort was expended by the user before they decided to give up?*
Question8: *When are they successful with a feature?*
Question9*: What learning barriers were faced?*
Question10: *What is the end-users' general impression of the software?*
Question11: *What are the levels of curiosity, motivation, frustration, time pressure, complexity of the task?*
Question12: *Do age, gender, experience, education background seem to have an impact on their usage of features?*

Level 3 and Level 4 Questions: *Questions asked of patterns across multiple cases and how they can be tied to theory (since we have not reached data analysis, this list is incomplete, and will grow.)*

Question1: *If there is an even distribution of gender, are the findings applicable across similar gender?*

Question2: *Is expertise an independent factor to the findings?*

Level 5 Questions: *The analysis and findings are necessary before we can complete this level of questions*

Question1: *Are inefficient strategies a major cause of non utilization or reluctance to use features?*

Potential Sources of information (Level2): The sources of information for the above questions can be obtained by observation, and interviews. The propositions will be a guide to what data could be used to complete our analysis.

Proposition1: exploratory learning, help of colleagues, internet, online forums, documentation, the embedded help system, or library [12].

Proposition2: May be tied to the help system [7], the time [3], self-efficacy [1], gender [2, 14], background (age, their education), curiosity [10], level of expertise, dire necessity to use a feature, frustration, motivation, interest (interest in learning, versus wanting to accomplish the task at hand), complexity of the task, resistance to adopt new features [11, 2], past negative experiences, poorly designed system etc.

Proposition4: Six learning barriers – Design, selection, coordination, use, understanding and information [9]

Guide for the case study report:

Outline: The outline of our report would be the following:

- Research question followed by the hypothesis.
- Description of the research design, apparatus and data collection procedure
- Presentation of the data collected
- Analysis of the data
- Discussion of the findings
- Conclusion

Format for the data: Our observations will be recorded in a tabular format, with a clear distinction on the raw data and our thoughts. Our interviews will be a collection of notes along with transcribed audio recordings. The questionnaire data will also be entered in a tabular format. Our notes on the documentation and the software itself will be a collection of notes.

Use and presentation of other documentation:

The software documentation is for us to understand the software, and also how effective the documentation served a beginner (namely the investigators). Using the beta version of the software will help clarify the questions or aspects of the software that we need to pay specific attention to when the participants use it. These documents will be attached to the appendix. We also analytically evaluate the software in terms of Cognitive Dimensions framework [8], and include the guidelines that have been followed.

Bibliographical information:
The references, the questionnaires used will be in the bibliography and the appendix.

Case Study Database

The following data will be collected:

- Direct observation of the end-user programmers
- Direct observation of the training session
- Questionnaires to the end-user programmers
- Documentation of the software (analytical evaluation)
- The beta version of the software (analytical evaluation of the software)
- Semi structured interviews with the end-user programmers

Direct Observation of End-User Programmers (DB_A)

Table 1: Chain of events – an overview/summary of events observed in chronological order (note down emotions, interruptions, comments from the participants, time taken for each event). (Example: Event1: Checking emails and their calendar, Event 2: making small talk with colleagues, lasting for 10 minutes, Event3: task1, lasting for 15 minutes, Event4: coffee break, etc). When the user starts working on a task, branch to table 2.

ID: DB_A1

What I observe	My thoughts
Event1:	
Event 2:	

Event n:	

Table2: Characteristics to be noted when the user starts working on a spreadsheet (BE ATTENTIVE to usage of EXCEL features Vs. AUDIT features. CLEARLY write down WHICH feature. MENTION the TASK ID while recording the following)

ID: DB_A2_X (substitute X with Task number)

What to collect: What I observe	My thoughts
Subject ID	
Task ID	
Time of the day	
Mood of the participant	
Disturbance/noise level	
Time of commencement of the task	
What is the task? (use an ID to refer to it later, example: task1) describe the events involved in the task:	
Level of complexity of the task (if you can judge it, or report it from the interview)	
Interest level/enthusiasm to get the task done (also include motivation)	
Interest level to learn a feature (inclusive of motivation) (include FEATURE used)	
Expression of lack of understanding of the feature/misinterpretation of the feature	
Persistence to use the feature	

Concentration while using the software (indications – how often do they check emails, browse the internet in the middle of working on their task etc)	
Any resistance to use a feature?	
Past negative experiences if they happen to mention	
Notes on the usage of help system, and documentation of the software	
Curiosity levels	
Frustration levels	
Amount of time spent on the task	
Any deadlines that they mentioned?	
Any rough count on the number of features that were used for the task?	
How many times were they 'stuck'? (give an ID… stuck1.1 refers to task1, stuck1, give a brief description, and refer to it later)	
Comments made by the participant during the task (if any about their day etc, which could be tied to other characteristics)	
Strategies adopted to clarify their understanding (exploratory learning, help of colleagues, internet, online forums, documentation, the embedded help system, or library/books) MENTION the 'STUCK id'	
Learning barriers (Design, selection, coordination, use, understanding and information) (associate it with STUCK id)	
Questions/surprises that you might like to address, during the interview	
Design guidelines that were useful (w.r.t Cognitive Dimensions) (mention TASK and STUCK id)	
Abstraction Gradient: Closeness of Mapping: Consistency: Diffuseness/Terseness:	

Error Proneness: Hard Mental Operations: Hidden Dependencies: Premature commitment: Progressive evaluation: Role Expressiveness: Secondary Notation and Escape from Formalism: Viscosity: resistance to local change Visibility and Juxtaposability:	

Direct Observation of the Training Session (ID: DB_B)

ID: DB_B1

What I observe	My thoughts
Time of the day	
Length of the session	
Number of trainers	
Was there a demo?	
Were there any handouts?	
Additional artifacts handed out (example whom to contact for help, etc)	
Questions asked by the attendees, along with the answers	
Were there any hands on training?	
Additional notes on presentation/training	

Questionnaires (ID: DB_C)

Background Questionnaire's data ID: DB_C1

Subject ID	Age	Gender	Educational background	Department (profession)	How many years of professional experience with Excel	How many classes in computer related applications (inclusive of programming)	How many days of experience with Audit
A							
B							
C							
D							

Self Efficacy Questionnaire's data ID: DB_C2

Subject ID	Pre-Self Efficacy Score	Post Self Efficacy score	Difference in Self Efficacy
A			
B			
C			
D			

Analytical Evaluation of the Audit product (ID: DB_D)

Evaluation of the Documentation ID: DB_D1

What to observe	My thoughts
How readable?	
Did it personally help you while trying to understand the software	
Does it cover all aspects of the software (from using features to error recovery)	
How structured is the document (logical flow)	
Were there sufficient examples?	
Were the examples non trivial and representative of problems commonly faced?	
Other Comments	

Evaluation of the software (primarily based on Cognitive Dimensions) ID: DB_D2

What to observe (with examples)	My thoughts
Abstraction Gradient:	
Closeness of Mapping:	
Consistency:	
Diffuseness/Terseness:	
Error Proneness:	
Hard Mental Operations:	
Hidden Dependencies:	
Premature commitment:	
Progressive evaluation:	
Role Expressiveness:	
Secondary Notation and Escape from Formalism: Viscosity: resistance to local change	
Visibility and Juxtaposability:	

Semi structured Interview with the end-user programmer (ID: DB_E)

Semi structured interview ID: DB_E1_X (substitute X with the subject id)

Question Asked	**Answer given**
1. How would you rate the tasks you performed today in terms of their complexity?	
2. Do you have any deadlines to complete the tasks you worked on today?	
3. What were the potential benefits you expected from the software that made you register for a training session?	
4. How useful do you think the Audit software was for your tasks?	
5. How useful was the training session to help you get started?	
6. How useful was the documentation?	
7. I noticed you using XXX features a lot. Were they helpful? Any reasons for liking them (may be follow up)	
8. Do you have any feature in Excel that you would not like to use? (follow up: Why, what are they, what would you use in substitution)	
9. Do you have any feature in Audit that you would not like to use? (follow up: Why, what are they, what would you use in substitution)	
Potential follow ups: if you note down any **learning barrier** that you would like to discuss with them, about some particular **problems they were stuck, their feelings** etc.	

References

[1] Bandura, A. Self efficacy: Toward a unifying theory of behavioral change. Psych. Review 84(2), 1977, 191-215.

[2] "Effectiveness of End-User Debugging Software Features: Are There Gender Issues?", Laura Beckwith, Margaret Burnett, Susan Wiedenbeck, Curtis Cook, Shraddha Sorte, Michelle Hastings, ACM Conference on Human Factors in Computing Systems, Portland, Oregon, April 2005.

[3] Blackwell, A. First steps in programming: A rationale for attention investment models, *IEEE Symp. Human-Centric Comp. Langs. Envs*., 2002, 2-10.

[4] Bhavnani, S. K., and John, B. E. From sufficient to efficient usage: An analysis of strategic knowledge. *ACM Conf.Human Factors in Computing Systems*, 1997, 91-98.

[5] Carroll, J. and Rosson, M. Paradox of the active user, In *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J. Carroll (Ed.), MIT Press, 1987.

[6] Carroll, J. M. (Ed.), *Minimalism Beyond the Nurnberg Funnel,* MIT Press, 1998.

[7] Dworman, G. and Rosenbaum, S. "Helping users to use help: improving interaction with help systems," in CHI 2004 Extended abstracts. Vienna, Austria: pp. 1717-1718

[8] Green, T., Cognitive dimensions of notations, In A. Sutcliffe and L. Macaulay (Eds.) People and Computers V. Cambridge University Press, 1989. pp 443-460.

[9] Ko, A., Myers, B., Aung, H., Six learning barriers in end-user programming systems, *IEEE Symp. Vis. Lang. Human-Centric Comp.,* 2004, 199-206.

[10] Lowenstein, G. The psychology of curiosity, Psycho-logical Bulletin, 116, 1, 1994, 75-98.

[11] The Influence of User Perceptions on Software Utilization: Application and Evaluation of a Theoretical Model of Technology Acceptance, Michael G. Morris and Andrew Dillon. IEEE Software, 14(4), 1997, 58-76.

[12] Reiman, J., A field study of exploratory learning strategies, ACM Transactions on Computer-Human Interaction, 1996, 189-218.

[13] Venkatesh, V. and Morris, M. Why don't men ever stop to ask for directions? Gender, social influence, and their role in technology acceptance and usage behavior, *MIS Quarterly,* 24(1), 2000, 115-139.

[14] Venkatesh, V. "Determinants of Perceived Ease of Use: Integrating Control, Intrinsic Motivation, and Emotion into the Technology Acceptance Model," *Information Systems Research* (11:4), 2000, pp. 342-365.